ESD-TR-68-248

Technical Note

1968-31

Numerical Integration of Differential Equations by Three Methods — Equations for, Description of Computer Routines, Discussion of Results

W. B. Smith

21 August 1968

Prepared under Electronic Systems Division Contract AF 19(628)-5167 by

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Lexington, Massachusetts

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-5167.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS BY THREE METHODS – EQUATIONS FOR, DESCRIPTION OF COMPUTER ROUTINES, DISCUSSION OF RESULTS

W. B. SMITH

Group 63

TECHNICAL NOTE 1968-31

21 AUGUST 1968

LEXINGTON MASSACHUSETTS

ABSTRACT

Three very different methods (from the literature) for the integration of differential equations by numerical methods are described. Formulas are given for two methods, and a computer program realization of the third is described.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

NUMERICAL INTEGRATION OF DIFFERENTIAL EQUATIONS BY THREE METHODS -- EQUATIONS FOR, DESCRIPTION OF COMPUTER ROUTINES, DISCUSSION OF RESULTS

I. INTRODUCTION

Our work requires the numerical integration of second-order differential equations to high accuracy (one part in 10^{11}, say) relatively close to the epoch, and good accuracy (1 part/10^7) over a long period. The methods used must also be efficient, lest many hours of computer time be wasted. To meet this need we have dealt with three distinct methods, which the present memo will describe in various detail for our own purposes and/or external utilization.

The potential user should be alerted that the methods to be described have been adapted to meet the particular needs of our program, and should not expect our specific realizations to be optimum for another kind of problem. This element of caution seems to be lacking in the literature, as is clarity. In fact, I developed our Adams-Moulton routine from no more than a pair of predictor-corrector formulas given in an unpublished memo as quotients of (finite-difference) operational polynomials. When -- after all the difficulties had been solved -- I found discussions of the method in the literature, not only did these discourses fail to help in refining our routine, but the difficulties we solved were not treated. Thus there is in this area ample opportunity to apply one's own insight and ingenuity.

II. METHOD OF ADAMS-MOULTON

This method rests upon a particular pair of quadrature formulas, and (unlike some methods; e.g., Runge-Kutta) steps from t_n to t_{n+1} with the aid of the solution at previous steps, say y_n, y_{n-1}, ... y_{n-p}; i.e., the method has "memory". Also, the predicted value y_{n+1} is "corrected" by a formula using y_{n+1}; hence, this is one of a class of "predictor-corrector" methods, a class which is apparently superior to all other methods.

Let us make the non-trivial supposition that the solution and its derivative are known at a sufficient number of previous points, and write

$$y_{n+1} \approx y_n + h \sum_{i=0}^{p} a_i \nabla^i y'_n$$

Here, the ∇'s are backward difference, with $\nabla^0 = 1$ by convention. We next use this estimate of y_{n+1} to obtain y'_{n+1} from the differential equation. Then we use a second quadrature formula involving y'_{n+1} to correct y_{n+1}:

$$y_{n+1} \approx y_n + h \sum_{i=0}^{p} b_i \nabla^i y'_{n+1}$$

If necessary, the second step may be repeated -- at the cost of a second computation of y'_{n+1}, and perhaps also the difference table.

The explicit form of the quadrature formulas have been succinctly derived by Glautz[1]. We give the formulas below and leave the derivation for the interested reader.

$$y_{n+1} = y_n + h \left[\frac{\nabla + \nabla^2 + \nabla^3 + \ldots}{\nabla + \frac{\nabla^2}{2} + \frac{\nabla^3}{3} + \ldots} \right] y'_n$$

$$y_{n+1} = y_n + h \left[\frac{\nabla}{\nabla + \frac{\nabla^2}{2} + \frac{\nabla^3}{3} + \ldots} \right] y'_{n+1}$$

One of the tasks in coding this method is that of finding the coefficients a_i and b_i up to high degree, a painful task if the polynomial division be done by hand. We have derived recursive formulae to perform the division which are well suited to machine computation.

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 1 - r_2 \\ a_2 &= 1 - r_3 - r_2 a_1 \\ &\vdots \\ a_p &= 1 - \sum_{\ell=1}^{p} r_{p-\ell+2}\, a_{\ell-1} \end{aligned}$$

where $r_\ell = 1/\ell$. And

$$
\begin{aligned}
b_o &= 1 \\
b_1 &= -r_2 \\
b_2 &= -r_3 - r_2\, b_1 \\
&\vdots \\
b_p &= -\sum_{\ell=1}^{p} r_{p-\ell+2}\, b_{\ell-1}
\end{aligned}
$$

It has heretofore been common to compute and print tables of such coefficients to save other workers the labor. However, a more current assumption is that everyone has a digital computer, for which the above formulas are as good as numbers.

It may be noticed that the prediction and correction formulae have been written in a form which appears to entail, computationally, the formation of difference tables of y', followed by multiplication and summation. If a second correction be required, then p more multiplications and additions would be necessary. However, the formulae can easily be rewritten so as to involve only the y', not its differences.

$$y_{n+1} = y_n + h \sum_{k=0}^{p} c_k\, y'_{n-k} \qquad \text{for prediction,}$$

where

$$c_k = \sum_{i=k}^{p} (-1)^k \binom{i}{k} a_i$$

if, by convention, $\binom{i}{o} = 1$.

$$y_{n+1} = y_n + h \sum_{k=0}^{p} d_k\, y'_{n+1-k} \qquad \text{for correction,}$$

where

$$d_k = \sum_{i=k}^{p} (-1)^k \binom{i}{k} b_i$$

The corresponding computer calculations are more efficient than from the difference notation. However, this approach has a defect which may or may not be serious for a given problem; i.e., the behavior of the higher differences is an excellent index for the accuracy and/or stability of the integration. It is common experience that increases in the higher derivatives of the "exact" solution, or errors in computer programming, may cause an increase in the higher differences of y', so that without some remedy the process becomes unstable. In our work, we accept the liability, and simply stop an integration when by some simple test an instability has obviously set in.

When used for integrating the second order differential equations of motion for a planet, the Adams-Moulton class of methods appears to have an inherent difficulty. Suppose that the orbit were an ellipse; then there would be four equations to be integrated in this method:

$$x_1' = x_2$$

$$y_1' = y_2$$

$$x_2' = \frac{-u\, x_1}{(x_1^2 + y_1^2)^{3/2}}$$

$$y_2' = \frac{-u\, y_2}{(x_1^2 + y_1^2)^{3/2}}$$

where (x_1, y_1) is the position and (x_2, y_2) the velocity. All such singly-integrating methods require double application as above to progress from acceleration to velocity to coordinate. But the peculiar form of the equations of motion for such methods suggests that our computer integrator could easily become a computer oscillator. Indeed, for a given number of differences in the predictor-corrector formulae, use of a step size greater than a corresponding threshold makes the system oscillate. We have found that a second correction alleviates the problem, at the cost of a second evaluation of driving

function. However, since the cure allows a step size larger by at least an order-of-magnitude, the saving in computation time is considerable. It is interesting that in the double-correction mode there seems to be no substantial difference between the first and second corrected solutions; this apparent paradox appears to illustrate the close relationship between accuracy and stability in numerical integration.

We have programmed the Adams-Moulton method and applied it with fine results to the N-body integration routine. With 11^{th} differences its performance as to accuracy/computation-time is far superior to that of the previous method due to Nordsieck (described below), with a computer time saving of about an order-of-magnitude.

Except for the restrictions of fixed step size and equal-interval tabular output the method has been programmed in a flexible, easily-used form for general use.

There are a number of techniques available for starting the Adams-Moulton method, which requires the solution at a sufficient number of points at the epoch to compute the difference tables. We simply use the Nordsieck routine, on which we can rely for solutions near the epoch with accuracy to 14 or 15 decimal places.

III. METHOD CALLED "ROYAL ROAD"

If formulas could be found by which one can doubly integrate in one stride directly from acceleration to coordinate, the just-discussed problem characteristic of single integration methods could be avoided. In fact such formulas exist, and we follow Kopal in referring to them as the Royal Road method, a description we prefer over JPL's name ("2^{nd} sum"). Since we have not yet quite finished programming the method, we shall limit the section primarily to a listing of formulas. We follow Hildebrand[2] in their derivation, but carry the algebra (for finding coefficients) in symbolic rather than numeric form, so as to have the computer find coefficients for high order terms. For correction,

$$\nabla^2 y_k = h^2 \left[\frac{\nabla}{-\log(1-\nabla)} \right]^2 y''_k$$

$$= h^2 \left[\sum_{i=0}^{\infty} b_i \nabla^i \right]^2 y''_k$$

with the b_i the same as in an earlier formula for the Adams-Moulton method. To carry out the squaring operation let $u_k = v_k = b_k$. Then,

$$\nabla^2 y_k = h^2 \sum_{i,j=0}^{\infty} u_i v_j \nabla^{i+j} y''_k$$

$$= h^2 [u_o v_o + (u_o v_1 + u_1 v_o) \nabla$$

$$+ (u_o v_2 + u_1 v_1 + u_2 v_o) \nabla^2 + \ldots] y''_k$$

$$= h^2 \sum_{i=0}^{\infty} w_i \nabla^i y''_k$$

$$w_i = \sum_{j=0}^{i} u_j v_i$$

For prediction, we operate on both sides with the "advancing" operator $E = 1/(1-\nabla)$:

$$E \nabla^2 y_k = \nabla^2 y_{k+1} = \frac{1}{1-\nabla} h^2 \sum_{i=0}^{\infty} w_i \nabla^i y''_k$$

$$= [1 + \nabla + \nabla^2 + \nabla^3 + \ldots] h^2 \sum_{i=0}^{\infty} w_i \nabla^i y''_k$$

$$= h^2 [w_o + (w_o + w_1) \nabla$$

$$+ (w_o + w_1 + w_2) \nabla^2 + \ldots] y''_k$$

$$= h^2 \sum_{i=o}^{\infty} e_i \nabla^i y''_k$$

$$e_i = \sum_{j=0}^{i} w_j$$

Since $\nabla^2 z_k = z_k - 2z_{k-1} + z_{k-2}$, these formulae become

$$y_{k+1} \approx -y_{k-1} + 2y_k + h^2 \sum_{i=0}^{p} e_i \nabla^i y''_k$$

$$y_{k+1} \approx -y_{k-1} + 2y_k + h^2 \sum_{i=0}^{p} w_i \nabla^i y''_{k+1}$$

where we have, for clarity, simply advanced the subscript of the correction formula originally given. Of course, these formulae can be expressed in terms of the y''_k and y''_{k+1} instead of their differences, as before.

The use of these royal road formulae in PEP raises a series of questions regarding computation of velocities. For our equations of motion the driving function y'' is a function of y' only in the relativity term, which is always numerically much smaller than other terms. In our equations for partial derivatives only the ∂/∂ (relativity factor) contains "velocity" terms, though of course these partials do not need to be as accurate as the coordinates. The velocities are also used in two other ways, both requiring considerable accuracy: (1) in processing "probe" (e.g., Mariner V) data the $\dot{r}$ must be calculated to better than 1 mm/sec, for which numerical differentiation of r may not be adequate; and (2) in restarting numerical integrations, where x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ are all needed to define the osculating orbit.

If the computation of velocities by integration of y'' could be avoided, a significant portion of computer time needed for PLANET and N BODY integrations could be saved. Hence, we shall incorporate the option of calculating velocities by numerical differentiation with only prediction of y'_{n+1}, since there will not be a second correction of y_{n+1}.

To derive the prediction formula we begin, as before, with the correction formula .

$$y'_n = -\frac{1}{h} \ln (1 - \nabla)\, y_n$$

$$= \frac{1}{h} \sum_{m=1}^{\infty} \frac{1}{m} \nabla^m y_n$$

Now apply E to both sides.

$$y'_{n+1} = E\, y'_n = \frac{1}{1-\nabla} \frac{1}{h} \sum_{m=1}^{\infty} \frac{1}{m} \nabla^m y_n$$

$$= (1 + \nabla + \nabla^2 + \nabla^3 + \ldots) \frac{1}{h} \sum \frac{1}{m} \nabla^m y_n$$

$$= [\nabla + (1 + \tfrac{1}{2}) \nabla^2 + (1 + \tfrac{1}{2} + \tfrac{1}{3}) \nabla^3 + \ldots] y_n.$$

To express these formulae in terms of tabular points, let

$$y'_n = \sum_{m=0}^{q} f_m \nabla^m y_n \quad ,$$

where

$$f_o = 0, \qquad f_m = 1/hm$$

Then, following the previous equations

$$y'_n = \sum_{k=0}^{q} q_k\, y_{n-k} \quad ,$$

where

$$q_k = \sum_{i=k}^{q} (-1)^k \binom{i}{k} f_i \quad ,$$

with analogous equations for the predictor.

For restarting in the absence of integrated velocities it will probably be possible to use velocities computed from the position and perhaps acceleration in the vicinity of the restarting time. We will give below a suggested formula for numerical differentiation of position; more elegant and accurate methods using the acceleration can certainly be derived, but may not yet be in the literature.

For accurate numerical differentiation we prefer some kind of central difference formula. It is easy to compute the mean of the previously given formula and the corresponding formula using forward differences.

$$\bar{y}'_n = \frac{1}{2}\left(y^{b}_{n}{}' + y^{f}_{n}{}' \right)$$

We imagine the previous formula to contain superscripts b (for backward differences), and write analogous formulas for forward differences:

$$y^{f}_{n}{}' = \sum_{m=0}^{q} f^{f}_{m} \, \Delta^{m} y_n \quad ,$$

where

$$f^{f}_{o} = 0 \, , \quad f^{f}_{m} = (-1)^{m+1}/hm$$

Since $\Delta^{s} y_n = \sum_{k=0}^{s} (-1)^{k} \binom{s}{k} y_{n+s-k}$, it follows in a not-so-simple way that

$$y^{f}_{n}{}' = \sum_{k=0}^{q} g^{f}_{k} \, y_{n+k} \quad ,$$

where

$$g^{f}_{k} = \sum_{i=k}^{q} (-1)^{i-k} \binom{i}{k} f^{f}_{i}$$

and hereby we can obtain $\bar{y}_n'$. We will discover by experiment whether it is accurate enough.

Since the Royal Road method will be stable without a second correction, and by using numerically differentiated velocities, we expect a saving in

computer time of better than a factor of 2, an important improvement for our projected very long integrations.

IV. METHOD OF NORDSIECK

The third method is due to A. Nordsieck, and is described in detail in Ref. 3. We have programmed and thoroughly checked out a double-precision "package" based on the reference. In this section we comment on the method and upon some aspects of realizing package.

The Nordsieck method is distinguished by two features: (1) it is self starting, and (2) the step size is automatically changed when needed. Since the method is one of the class of predictor-corrector methods, which involve the "memory" of the solution for earlier values of the independent variable (t), starting is naturally a problem because in the beginning there is nothing to remember! The starting procedure in the Nordsieck method is so accurate and easy to use that the package is very useful as a starting routine for other methods, and is so used. The importance of a good starting procedure should be emphasized; the "inherited error" due to inaccuracy in computing the first few values of a solution can easily dominate subsequent errors.

The variable-step-size feature performs as follows: the step size (h) is such that the equation is integrated over a unit distance (from $t = t_1$ to $t = t_1 + 1$) with an accuracy not worse than a specified value (the program input variable named EPS). Changes in h are accomplished efficiently and stably, so that even poorly behaved solutions, such as (continuous) stepwise-linear functions, are efficiently found. In the Planetary Ephemeris Program (PEP) work the method proved to be especially useful for integrating the equations of motion of the asteroid Icarus, which, because of its high eccentricity and near-collision with the Earth, required the diminution of h by about a factor of 10 at perihelion and at close approach.

Analogous with the utility of observing the high order differences in other methods, the behavior of h provides an index to the current nature of the differential equation. In particular, in a number of instances the onset of a rapidly decreasing h has led to the discovery of programming errors outside the integrating package.

One fairly subtle point should be mentioned for the benefit of those who may wish to use our package for the Nordsieck method. Ordinarily no loss of accuracy is entailed when h abruptly diminishes; indeed, it usually diminishes precisely to retain accuracy. An exception occurs when h is required temporarily to decrease by more than a factor of 2 in order to compute a solution at a preset tabular point. We prevented this from occurring at any tabular point after the second (excluding the epoch) by allowing h to double only at any other tabular point, via logic involving L5. However, the user must select the initial interval (HC) and/or the tabular interval (HMX) so that for his problem this circumstance will not occur at the first or second tabular point.

Because of difficult algebraic problems involved in deriving certain coefficients, and because of questions about the stability of the variable h, the method has been programmed only for the 5^{th} and 6^{th} degrees. While satisfactory for general use, it turns out that for our work in planetary orbits other methods of higher degree are much more efficient -- with the probable exception of Icarus and other bodies with similar orbits. When the higher degree differences of a solution are large for a given h, the Nordsieck method responds by diminishing h, whereas a method which carries along these higher differences can condone the original h without loss of accuracy. It may happen that the increased computation time for high-degree methods (for a given h) may be more than offset by the larger allowable value of h, especially when evaluating the right-hand side of the differential equation(s) requires a large share of the gross computation time. Hence, we have found that for the integration of the 9-body problem the Nordwieck method is more consumptive of computer time by a factor of (roughly) 10 in comparison with the Adams-Moulton method, for the given required accuracy.

Some additional comment is needed regarding the criteria for allowing h to double. These criteria were carefully tailored to be efficient for our kind of celestial mechanics integrations, and may or may not be suitable for other work. A note of caution should be offered to anyone who might wish to change them. We have found that a single step at a doubled interval when non-optimum

criteria were used could produce a truncation error, propagated through subsequent steps, which dominated all later errors. That is, with no saving of time for the one step, accuracy was nevertheless greatly diminished. With proper selection of criteria this does not occur.

REFERENCES

1. R. D. Glautz, Computing Services Report #13, Aerojet-General Corp. (May 1960).

2. F. B. Hildebrand, "Introduction to Numerical Analysis "(McGraw-Hill, 1956), Chapter 5.

3. A. Nordsieck, Math. of Comp., p. 22 (January 1962).

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None

3. REPORT TITLE

Numerical Integration of Differential Equations by Three Methods – Equations for, Description of Computer Routines, Discussion of Results

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Technical Note

5. AUTHOR(S) *(Last name, first name, initial)*

Smith, William B.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
21 August 1968	18	3

8a. CONTRACT OR GRANT NO. AF 19(628)-5167 b. PROJECT NO. 649L c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Note 1968-31 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-68-248

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Air Force Systems Command, USAF

13. ABSTRACT

Three very different methods (from the literature) for the integration of differential equations by numerical methods are described. Formulas are given for two methods, and a computer program realization of the third is described.

14. KEY WORDS

differential equations | numerical integration | computer routines